Scheduling Working Group
Scheduling Working Document: 10.4
Category: Informational

Schwiegelshohn/Yahyapour
Computer Engineering Institute

August 2001

# Attributes for Communication between Scheduling Instances

## Status of this Draft

This draft provides information for the grid scheduling community. Distribution of this memo is unlimited.

## Copyright Notice

## Abstract:

This document describes a set of attributes of a lower level scheduling instance – as found locally on systems – that can be used by a higher level scheduling instance – as found in an Grid environment to interact with remote local scheduling systems. This set of terms provides directions for implementers of new schedulers that are used in computational grids.

---

## 1. Introduction

A Computational Grid typically consists of a variety of different resources with different owners. Usually those resources are not exclusively dedicated to Grid usage. For instance, a computer may temporarily be removed from the grid to solely work on a local problem. Many of those single Grid resources use local management systems that may include local scheduling instances.

In general, sites can freely participate in Grid computing by offering resources provided that certain conditions like security requirements are met. The interaction between those grid resources during the execution of a job requires a scheduling layer that uses a different scheduling paradigm than that of local or centralized schedulers. While local scheduling usually involves a single scheduling instance that has access to all system information, grid scheduling requires interaction to remote sites and their local scheduling systems. This suggests the use of several scheduling layers for the grid. Although the details of this scheduling architecture has not yet been decided, it is clear that those layers need to exchange information.

In this context, a distinction is made between ***lower level scheduling instances*** for the local scheduling of resources and ***higher level scheduling instances***, that are used for interaction in coordinated scheduling in a Computational Grid. Both scheduling instances need to work efficiently together in order to make best use of the Grid resources. This does not imply that the scheduling instances in a Grid must form a static hierarchy. We use the expression *higher level scheduler* to denote a scheduler that is actually querying another scheduler for possible allocations for a query. This other scheduler is called a *lower level scheduler*. Therefore, the expression “higher” results from the direction a query takes during the process of scheduling and does not indicate a fixed hierarchy of schedulers. Hence, a higher level scheduler at one moment may as well be a lower level scheduler next during another scheduler communication. Those lower level schedulers may be part of queuing systems, like PBS, LoadLeveler, LSF, NQS, or they may already include scheduling features as those provided by GARA in Globus.

The purpose of this document is the definition of attributes that describe those available features of such a lower level scheduling instance that can be exploited by a higher level scheduling instance. These attributes facilitate the interaction and cooperation between the different levels of Grid scheduling system as there may be different local schedulers with different features. Note that the document only focuses on  the attributes for scheduler communication without addressing mechanisms for this communication.

Also these attributes do not describe the structure and syntax for the resource description, like, for instance, the minimum number of processors or the amount of memory requested. This information can be accessed through the Grid information service. While this is also an important part of the grid infrastructure it is specified in a different document. Instead only the features offered by the lower level scheduling instance to the higher level scheduling instance are addressed in this document. Consequently, there are no attributes in this document to determine the set of resources a lower level scheduler is responsible for. Further, although this document does not define the interface between the different scheduling layers in detail, it can be seen as the first step in this direction.

Note, that the features described by the attributes are neither an obligation nor a limitation for the design of a lower level scheduling system. For instance, some features are not relevant for certain resource types. Therefore, they are not considered in the corresponding local schedulers. But the attributes defined in this document are restricted to be used in connection with a specific resource type.

Finally, this list focuses on scheduling systems used today. Future developments will provide additional features supported by the scheduling instances that will require corresponding adaptation of the scheduling attributes. Therefore, this list should serve as a starting point.

## 2. Typical Scenario

In a typical example, the higher level scheduling instance (a grid job scheduler) coordinates the scheduling for multi-site application or helps to select the best resources for a job among different possible resource offers. Typically, this scheduler itself has no direct control over resources. Therefore, it needs to communicate with and appropriately trigger lower level scheduling instances. Those lower level schedulers either control resources directly or have some kind of access to their local resources. However, note that the concept is not restricted to two levels of scheduling instances. Therefore the lower level scheduling instance can be a local scheduler for a single resource or it may be a scheduling system that manages several resources. If more than two levels of scheduling instances are used then a scheduling instance in the middle needs to collect attributes from possibly several lower level scheduling instances, combine them and provide those combined attributes to a higher level scheduling instance. In this document we do not address the method of combining attributes, as there are several possible solutions to this problem, like a logical and or a attributes associated with a list of resources. The type of the resources handled by lower level schedulers is not restricted to computing resources but may also include other resources as, for example, some network bandwidth that is controlled by a bandwidth broker.

In this document we will use the term “allocation” for assignments of resources to a request. An allocation is *tentative* until it is finally executed, that is, until resources are actually consumed. The schedule maintained by some management instance gives information on the planned or guaranteed allocations. Also note that any guarantee of an allocation only refers to the guaranteed assignment of resources, but does not guarantee the completion of a job.

As a simple example assume a Computational grid that includes the following resources from different institutions:

- Several clusters of workstations
- a visualization cave
- a special database
- a network with bandwidth brokerage that connects the other resources.

On request of an application a grid job scheduler tries to find a combined allocation that includes a set of 5 workstations, a visualization cave during one stage and the access to a database at another stage. To this end, the grid job scheduler has to interact with the local management instances of the various resources to find suitable allocations. In order to do its job the grid job scheduler needs not only information about the available resources but also about the available features of the corresponding local schedulers.

For instance, the grid scheduler wants a guaranteed completion time of allocation for one stage in order to make an advanced reservation for the visualization cave in the next stage. If such a guarantee cannot be provided by any of the corresponding lower level scheduling instances, the grid scheduler prefers an exclusive allocation to limit the influence of other independent jobs. Finally, if such a feature is also not available, the grid scheduler may reserve additional resources if it can preempt the job a one stage and migrate it to a less heavily used set of resources, if necessary.

We suggest that this information about the features of lower level scheduling instances can be provided in form of a list of attributes.

## 3. Attributes of allocation properties

These attributes are useful for a higher level scheduling instance to determine timing, lengths and reliability of allocations.

### 3.1 Revocation of an allocation

This attribute indicates that the local management may revoke an already made allocation while the allocation is still tentative. This may even happen although the higher level scheduling instance has fulfilled all its requirements to keep the allocation.  For instance, the resources of the allocation may be withdrawn from Grid usage or may be used to execute another job with a higher priority.  Therefore, this attribute indicates that the allocation is not guaranteed. Note that this attribute is independent of any process that must be executed by the user system or the higher level scheduling system to prevent deallocation according to the deallocation policy of a lower level scheduling system.

### 3.2 Guaranteed completion time of allocations

This attribute indicates that the local management system guarantees the completion time of an allocation. This does not necessarily mean that the actual allocation is known but only that the system guarantees that the requested resource allocation will be executed before a given deadline.

### 3.3 Allocations run-to-completion

This attribute indicates that the local management will not preempt, stop or halt an application after it has been started. The allocation will stay active on the given resources until the end of the requested time or the completion of the job.

### 3.4 Exclusive allocations

This attribute indicates that the allocation runs exclusively on the provided set of resources. The resources are not time-shared and the executed allocation is not affected by the execution and resource consumption of another allocation running concurrently.

### 3.5 Malleable allocations

This attribute indicates that the resource-set of an application can change the resource set during execution of the allocation, that is, the local management system supports the addition or removal of resources from applications during run-time. This modification of the allocation does not require run-time involvement of the higher level scheduling instance.

Option: Moldable allocations

This attribute indicates that the local resource management can only increase the resource set of an allocation during run-time. In contrast to malleable allocations resources are not taken from the application.

## 4. Attributes of available information

This attributes allow the higher level scheduling instances to determine which information it may obtain from a lower level scheduling instance and how reliable this information is.

### 4.1 Access to the tentative schedule

This attribute indicates that the local management returns on request a current schedule for present and future allocations. This information can be helpful for the grid job scheduler to determine suitable timeslots that can be used, for instance, for co-scheduling in multi-site computing.
There are many options possible for this attribute.

Option: Projected start time of a given allocation

Option: Partial information on the current schedule.

The access to the schedule may be limited due to system policies or restrictions of the local management.

### 4.2 Exclusive Control

This attribute indicates that the lower level scheduler is in exclusive control of the resources and no allocations can be scheduled on those resources without using this lower level scheduling instance. This information is useful to determine the reliability of scheduling information.

### 4.3 Event Notification

This attribute indicates that the lower level scheduling instance supports event subscription. A higher level scheduling instance may subscribe to specific events and use this information for monitoring or rescheduling of allocations.
Note that this document neither specifies the event types nor gives a definition of interfaces to query supported event types. This attribute only indicates that such a mechanism is available.

## 5. Attributes for manipulating the allocation execution

These attributes indicate supported scheduling functionalities of the lower level scheduling instance that can be used by a grid job scheduler to directly modify a running allocation.

### 5.1 Preemption

This attribute indicates that the local resource management allows the temporary preemption of an allocation by a higher level scheduling instance. In this case the corresponding application is stopped but remains resident on the allocated resources and can be resumed later. This preemption is not synonymous with the preemption in a multi-tasking system that typically happens in the time range of milliseconds. It only indicates that the local management offers the ability to remotely initiate a preemption of another allocation to e.g. temporary free resources for other usage or to synchronize two allocations on different resources.

### 5.2 Migration

This attribute indicates that the local management system allows the migration of an application or part of an application from one resource set or subset to another set. This way an application can be stopped at one location, the corresponding data moved to another location and the execution can be resumed.

### 5.3 Checkpointing

This attribute indicates that the local management system supports the checkpointing of a job. A file of a checkpointed job is generated that allows a later continuation from that point. The checkpoint file may also be migratable under some conditions.

## 6. Attributes for requesting resources

These attributes indicate functionality of the lower level scheduling instance that are useful for a higher level scheduling instance to determine which information must be provided when requesting resources and which answers to expect.

### 6.1 Allocation offers

This attribute indicates that the local resource management supports the generation of potential resource allocations for a request.
For instance, if several resources are capable to fulfill a request, a grid job scheduler can first query those systems for the allocation and afterwards make its decision to accept the allocation.

Options: Single/multiple offers for a request.

This option indicates that the local management system can provide several offers for a request with possible overlapping allocations.
For instance, a grid job scheduler may use this feature for multi-site application where corresponding allocations must be found on different sites.

### 6.2 Allocation cost/objective information

This attribute indicates that the local management system can return objective or cost information for an allocation.
In case of several allocation offers, a grid job scheduler can, for instance, use this information for the evaluation.
The cost/quote for a specified allocation usually relates to the policy that is applied by the lower level scheduling instance. This represents the scheduling objective of the owner of the resource.

### 6.3 Advanced reservation

This attribute indicates that advanced reservation is supported according to the proposed advanced reservation protocol.

### 6.4 Requirement for providing maximum allocation length in advance (Support/No Support for undetermined allocation length)

This attribute indicates that the local management system requires an allocation length to be given in advance. Historically, resource requests often have been submitted without additional information on the amount of time that the resources will be used. These programs have been started and run until completion. Current scheduling algorithms as, for example, the backfilling algorithm require additional information on the maximum allocation length.

## 5.4 Deallocation policy

This attribute indicates that some kind of deallocation policies for pending allocation applies. Some systems pose requirements that must be met to keep the allocation valid.
An example for such a policy is the requirement that an allocation must be repeatedly confirmed until the execution of the allocation.
These policies must be further specified to allow the higher level scheduler to keep the allocations.

## 6.5 Remote co-Scheduling

This attribute indicates that the local management instance allows co-scheduling where the actual resource allocation and schedule is generated by a higher scheduling instance. This includes the generation/cancellation of allocations on the local schedule by the higher level scheduler..
For instance, a grid job scheduler can use this property to quickly co-allocate resources at different sites in order to fulfill a more complex job request.

## 6.6 Consideration of job dependencies

This attribute indicates that the lower level scheduler takes dependencies between allocations into account if there are provided by the higher level scheduling instance. For instance, in case of a complex job request the lower level-scheduling will not start an allocation if the completion of another allocation is required and still pending.

## 7. Authors' Address

Uwe Schwiegelshohn
Uwe.Schwiegelshohn@uni-dortmund.de

Ramin Yahyapour
Ramin.Yahyapour@uni-dortmund.de

Computer Engineering Institute, University Dortmund